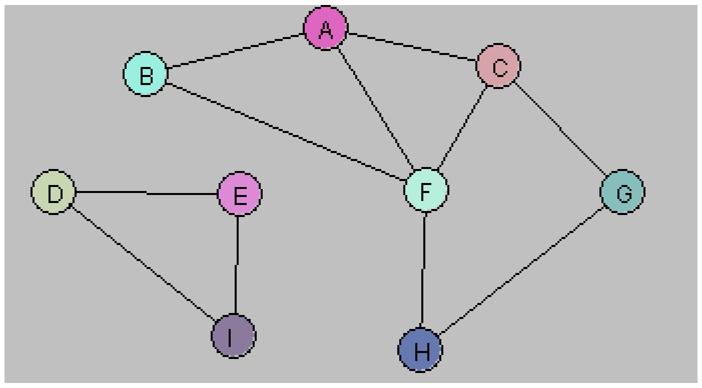
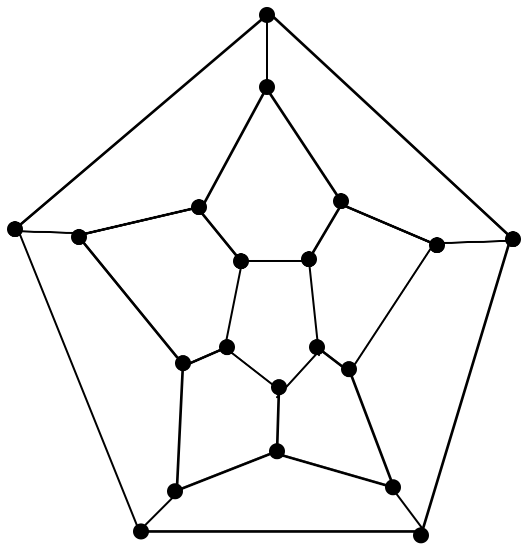
Lab 11

1. Answer questions about the graph G = (V,E) displayed below.



   A. Is the graph G connected? If not, what are the connected components for G?
   B. Draw a spanning tree/forest for G.
   C. Is G a Hamiltonian graph?
   D. Is there a Vertex Cover of size less than or equal to 5 for G? If so, what is the Vertex Cover?

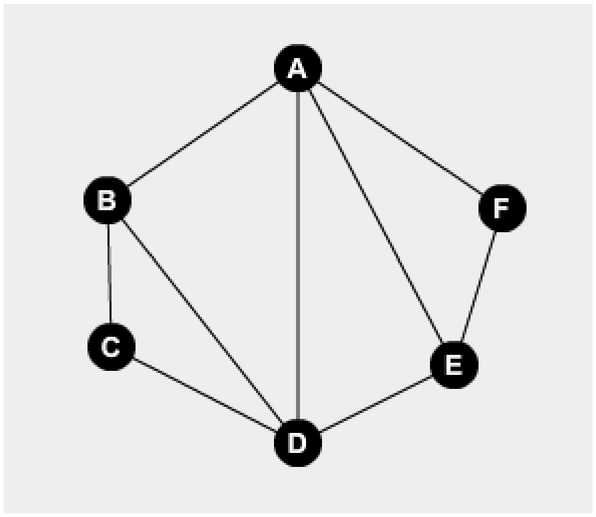
2. *Hamiltonian Graphs.* The following graph has a Hamiltonian cycle. Find it.



3. *Vertex Covers.* Create an algorithm for computing the smallest size of a vertex cover for a graph. The input of your algorithm is a set V of vertices along with a set E of edges. Assume you have the following functions available (no need to implement these):
   - computeEndpoints(edge) – returns the vertices that are at the endpoints of the input edge
   - belongsTo(vertex, set) – returns true if the input vertex is a member of the given set

   Hint: Loop through all subsets of V. For each subset W, check to see if W is a vertex cover. Do this by looping through all edges; for each edge e, check to see if at least one of its endpoints lies in W.

4. Compute two spanning trees for the graphs below using algorithms we discuss in class. (You can start with vertex A) Are the two spanning trees same?



5. Write the pesudo-code for compute connected components algorithm discussed in class. Your algorithm can be built on top of DFS discussed in the slides.

   Hint:
   Make a ConnectedComponentSearch subclass of DFS

   Initialize ArrayList<List<Vertex>> componentMap;
   Initialize HashMap<Vertex,Integer> vertexComponentMap; //for other tasks…
   CurrentComponentNumber ← 0

   Algorithm: additionalProcessing

Algorithm: processVertex(v)

Algorithm: computeConnectedComponents

6. Write the pesudo-code for the algorithm, discussed in class, that computes the shortest path length between two vertices in a graph. You can assume that:
    a. The graph is connected.
    b. A version of BFS is provided that accepts a specified starting vertex.

    Hint:
    Make ShortestPath a subclass of BFS

    Initialize HashMap<Vertex, Integer> levelsMap
    Initialize HashMap<Vertex, Vertex> parentMap

    Algorithm processEdge(Vertex v, Vertex w)

    Algorithm processVertex(Vertex v)

    Algorithm computeShortestPathLength(Vertex s, Vertex v)